

Documento de entrega (Numerales 1 y 2 (inciso a.) documento)

Entrenamientos forajidos o mixtos.

En este documento, se muestra como realizar entrenamientos en dos casos específicos:

1) Cuando se requiere entrenar un modelo por fuera de la plataforma de entrenamientos AITuring.

Todos los entrenamientos en la plataforma o externos utilizan Tensorflow y Docker.

Pasos:

A) para entrenamientos por fuera de la plataforma se utilizan 4 maquinas disponibles:

* [efficientdet-training-2, efficientdet-training-3, johnnew-efficientdet y posters-training.](#)

En las maquinas mencionadas, se encuentra todo lo necesario para realizar entrenamientos fuera de la plataforma.

B) La configuración de los entrenamientos se realiza con un archivo que contiene los siguientes parámetros (ejemplo, parámetros son indicativos y cambian con el entrenamiento), este archivo puede ser ubicado en cualquier lugar de la maquina:

```
URL_DATASET=https://ai-trade.s3.amazonaws.com/dataset_output/NUEVO-DATASET-PRODUCTO-GONDOLAS_MONDELEZ%20%20CHILE_VERSION-72021-06-18-16-03.zip
BACKBONE_REF=efficientdet-d2
NUM_EPOCHS=100
BATCH_SIZE=4
MODEL_CHKPTS=efficientdet-d2-dataset-MON-CHILE-GONDOLAS-PROPIOS-2021-18-06
```

Específicamente, cada uno de los parámetros se detalla a continuación:

- * **URL_DATASET:** Path https en s3 donde se encuentra el dataset con el cual se desea entrenar (string).
- * **BACKBONE_REF:** Actualmente existen tres posibilidades para este parámetro (string), efficientdet-d1, efficientdet-d2 y efficientdet-d3. Experimentos han mostrado que efficientdet-d3 da buenos resultados siempre y cuando se cuente con un dataset de tamaño considerable (mas de 20000 imágenes), pero aconsejamos solo utilizar la versión 1 (en total uso en Abi) y 2 (la cual es utilizada en su mayoría para Mondelez). La elección de una versión u otra no sigue un estándar particular y se debe elegir basado en la experiencia.
- * **NUM_EPOCHS:** Numero de vueltas (integer) que dará el entrenamiento. Ese parámetro se selecciona dependiendo de la tarea a resolver, para POP se asignan 20 vueltas, para botellas y productos en general, son 100 vueltas.
- * **BATCH_SIZE:** (integer) Numero de imágenes que toma la GPU durante una iteración del entrenamiento. Este parámetro es muy dependiente de la capacidad en memoria de la GPU, pero en grandes líneas si se utiliza efficientdet-1 este parámetro se coloca en 6, si se utiliza la versión 2 se coloca en 4 y finalmente si se utiliza la versión 3 se coloca en 2.
- * **MODEL_CHKPTS:** (string) Nombre final del modelo de salida, la carpeta de salida será, /home/user/semillas/MODEL_CHKPTS. El modelo final para producción /home/user/semillas/MODEL_CHKPTS_freeze.

C) Luego de haber creado el archivo en el punto B, se puede lanzar el entrenamiento con el siguiente comando.

```
sudo docker run --rm -v /path/salida/modelos:/workspace/  
saved_checkpoints --shm-size=8g --ulimit memlock=-1 --  
ulimit stack=67108864 --gpus all --env-file  
archivo_creado_punto_B.list training image
```

En este punto hay que considerar los recursos de la maquina, el fallo que se puede presentar en general de este comando, es por falta de recursos disco duro de la maquina.

D) Finalmente los modelos entrenados se encontraran en la carpeta /path/salida/modelos, en esa carpeta es necesario comprimir cómo tar.gz la carpeta que termina con “_freeze” que es la que finalmente se despliega en operaciones.

2) Cuando la operación requiere combinar dos set de datos (datasets) en uno para luego ser entrenado por fuera de la plataforma.

Si la operación requiere combinar dos o mas datasets, es necesario hacer este proceso manualmente en un solo archivo .zip conservando la estructura de datos que tienen los datasets. Finalmente, se sube el dataset a un espacio publico en s3 y luego se siguen los pasos mostrados en el numeral 1.

ANOTACIONES GENERALES

1)

El código utilizado en estas maquinas es el siguiente:

https://git-codecommit.us-east-1.amazonaws.com/v1/repos/aituring_efficientdet

Para construir la imagen de Docker (training_image) utilizar :

```
$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/aituring_efficientdet  
$ cd aituring_efficientdet  
$ docker build training_image .
```

2)

Existe ya disponible una versión de este código estilo keras sin sesiones. Los parámetros de entrada son los mismos que en la versión por sesiones, lo único que cambia es la manera de realizar la inferencia. El código lo pueden encontrar en el siguiente enlace:

https://git-codecommit.us-east-1.amazonaws.com/v1/repos/aituring_dockertraining_keras

Para construir la imagen se utilizan los siguientes comandos:

```
$ git clone --recurse-submodules https://git-codecommit.us-east-1.amazonaws.com/v1/repos/  
aituring_dockertraining_keras
```

```
$ cd aituring_dockertraining_keras  
$ docker build training_image_keras .
```

Esta misma imagen puede ser utilizada para entrenar modelos con los mismos parámetros mencionados en el numeral 1, pero para ser utilizados en modo keras. Un ejemplo de como realizar inferencia con keras se puede encontrar en :

[aituring_dockertraining_keras/test_inference_aituring.py](#)

En el repositorio clonado anteriormente.